

**Govt. of Karnataka, Department of Technical Education**

Diploma in Computer Science & Engineering

**Fourth Semester**

**Subject: SOFTWARE ENGINEERING**

**Contact Hrs / week: 4**

**Total hrs: 64**

**Table of Contents**

<b>Chapter No</b>	<b>Topics</b>	<b>Hours</b>	<b>Marks</b>
1	The Software Problem	02	05
2	Software Process	10	26
3	Software Requirements Analysis and Specifications	08	20
4	Software Architecture	06	16
5	Planning a Software Project	06	16
6	Design	10	26
7	Coding	06	16
8	Testing	08	20
	Seminars and Guest Lectures from industry and institutes	05	
	Test	03	
	<b>Total</b>	<b>64</b>	<b>145</b>

**Detailed Contents**

1		<b>The Software Problem</b>
	1.1	Cost, Schedule and quality
	1.2	Scale and Change
2		<b>Software Processes</b>
	2.1	Process and Project
	2.2	Component Software Processes
	2.3	Software development Process Models
	2.3.1	Waterfall
	2.3.2	Prototyping
	2.3.3	Iterative development
	2.3.4	Rational Unified Process
	2.3.5	Time boxing Model
	2.3.6	Extreme programming and Agile processes
	2.3.7	Using process model in a project
	2.4	Project Management Process

3			<b>Software Requirements Analysis and Specifications</b>
	3.1		Value of good SRS
	3.2		Requirement process
	3.3		Requirement Specification
		3.3.1	Desirable Characteristics of an SRS
		3.3.2	Components of an SRS
		3.3.3	Structure of a Requirements Documents
	3.4		Functional Specification with use cases
			Basics
			Examples
			Extensions
			Developing use cases
	3.5		Other Approaches for Analysis
			Data Flow Diagram
			Entity Relationship Diagrams
	3.6		Validation
4			<b>Software Architecture</b>
	4.1		Role of Software Architecture
	4.2		Architecture Views
	4.3		Component and Connector view
		4.3.1	Components
		4.3.2	Connectors
		4.3.3	Example
	4.4		Architecture styles for C&C View
		4.4.1	Pipe and Filter
		4.4.2	Shared data Style
		4.4.3	Client Server style
		4.4.4	Some other styles
	4.5		Documenting Architecture Design
	4.6		Evaluating Architectures
5			<b>Planning a Software Project</b>
	5.1		Effort Estimation
		5.1.1.	Top-Down Estimation Approach
		5.1.2	Bottom-Up Estimation Approach
	5.2		Project Schedule and Staffing
	5.3		Quality Planning
	5.4		Risk Management Planning
		5.4.1	Risk Management Concepts
		5.4.2	Risk Assessment
		5.4.3	Risk Control
		5.4.4	A Practical Risk Management Approach
	5.5		Project Monitoring Plan
		5.5.1	Measurements

		5.5.2	Project Monitoring and tracking
	5.6		Detailed Scheduling
6			<b>Design</b>
	6.1		Design Concepts
		6.1.1	Coupling
		6.1.2	Cohesion
		6.1.3	The Open-Closed Principle
	6.2		Function Oriented Design
		6.2.1	Structure Charts
		6.2.2	Structured Design Methodology
		6.2.3	Example
	6.3		Object Oriented Design
		6.3.1	OO Concepts
		6.3.2	Unified Modeling language (UML )
		6.3.3	A Design Methodology
		6.3.4	Examples
	6.4		Detailed Design
		6.4.1	Logic/Algorithm Design
		6.4.2	State Modeling of Classes
	6.5		Verification
7			<b>Coding</b>
	7.1		Programming Principles and Guidelines
		7.1.1	Structured Programming
		7.1.2	Information Hiding
		7.1.3	Some Programming Practices
		7.1.4	Coding Standards
	7.2		Incrementally Developing Code
		7.2.1	An incremental coding process
		7.2.2	Test Driven development
		7.2.3	Pair Programming
	7.3		Managing Evolving Code
		7.3.1	Source Code Control and Build
		7.3.2	Refactoring
	7.4		Unit Testing
		7.4.1	Testing procedural units
		7.4.2	Unit testing of Classes
	7.5		Code Inspection
		7.5.1	Planning
		7.5.2	Self review
		7.5.3	Group review meeting
8			<b>Testing</b>
	8.1		Testing Concept

		8.1.1	Error, Fault and Failure
		8.1.2	Test Case, Test Suite and Test Harness
		8.1.3	Psychology of Testing
		8.1.4	Levels of Testing
	8.2		Testing Process
		8.2.1	Test Plan
		8.2.2	Test Case Design
		8.2.3	Test Case Execution
	8.3		Black Box Testing
		8.3.1	Equivalence Class Partitioning
		8.3.2	Boundary value Analysis
		8.3.3	Pair Wise Testing
		8.3.4	Special Cases
		8.3.5	State based testing
	8.4		White Box testing
		8.4.1	Control Flow Based Criteria
		8.4.2	Test Case Generation and Tool Support

General Objectives :

1	Know how costs, schedule and quality drive a software project
2	Learn the role of software process and a process model in a project
3	Understand the role of SRS in a project and how requirements are validated
4	Understand the software architectural views
5	Learn planning and estimation of a software project
6	Know the key design concepts of software engineering
7	Learn the structured code inspection process
8	Learn how testing is planned and testing done

**Specific Objectives:**

1	<b>The Software Problem</b>
	Differentiate between a student software and an industrial-strength software
	Learn the software quality attributes
	Learn the additional software quality attributes : Scale and Change
2	<b>Software Processes</b>
	Define Process, Project, Process Model
	Learn the components of software processes
	Explain different Software Development Process Models
	Illustrate temporal relationship between development and management process
3	<b>Software Requirement Analysis and Specification</b>
	Learn the value of Good SRS
	Explain the tasks involved in the requirement process

	Appraise the characteristics of a SRS
	Explain the components of SRS
	Describe the general structure of SRS
	Learn the functional specification with use cases with examples
	Learn the levels of abstraction used for developing Use cases
	Explain DFD , ERD
	Learn how a SRS should be validated
4	<b>Software Architecture</b>
	Appraise the role of software architecture
	Discuss architectural views
	Explain component and connector view
	Learn various Architecture Styles for C&C view
	Learn Documenting Architecture Design and Evaluation
5	<b>Planning a Software Project</b>
	Learn how to estimate effort and schedule for the project to establish project goals and mile stones and team size
	How to establish quality goals for the project and prepare a quality plan
	Identify high-priority risk that can threaten the success of the project and plan for their mitigation
	How to plan for monitoring a project using measurements to check if a project is progressing as per plan
	How to develop a detailed task schedule for the overall estimates and planning task
6	<b>Design</b>
	Identify the key design concepts
	Learn the structure chart notation for expressing the structure of a function oriented system
	Learn key concepts related to OO and UML that can be used to express an OOD
	Learn the guidelines for making a detailed design
	Learn to verify a design
7	<b>Coding</b>
	Know the programming principles and guidelines
	Appraise coding standards
	Learn the effective processes that developers use for incremental developing code
	Explain how to manage evolving code by using proper source code control and refactoring
	Explain unit Testing
	Learn the process of code inspection
8	<b>Testing</b>
	Learn the concepts and definition relating to testing
	Appraise how testing is planned and how unit test is done
	Learn to test case selection using black box and white box testing approaches

Textbook:

1. Software Engineering –A Precise Approach, Pankaj Jalote, edition 2010, Wiley India, ISBN: 9788126523115.

References:

1. Software Engineering A Practitioners Approach, Roger S Pressman, 2010, Tata McGraw Hill Publishing Co Ltd, ISBN 9780070701137
2. Software Engineering, Sajan Mathew, 2009 Reprint, S Chand publications

**Govt. of Karnataka, Department of Technical Education**  
**Diploma in Computer Science & Engineering**  
**Fourth Semester**  
**Subject: Software Engineering**

Max. Marks: 100

Max. Time: 3 Hours

Model Question Paper

- Note: 1. Section –I is compulsory.  
2. Answer any TWO questions from each remaining Sections.

Section - I

- 1 a. Fill in the blanks 5x1=5
- i. Once the software is delivered and deployed it enters \_\_\_\_\_ phase
- ii. \_\_\_\_\_ is a critical activity in software development
- iii. \_\_\_\_\_ are commonly used during problem analysis
- iv. A \_\_\_\_\_ connects a client to a server.
- v. A testing frame work is also called as \_\_\_\_\_
- b. Write a short note on white box testing 5

Section – II

2. a) Define Software quality attributes 5  
b) Explain with a neat diagram the components of software processes 10
- 3 a) Explain any one software development process mode? 10  
b) Which is more important – the process or the product? Why? 5
- 4 a) Explain the desirable Characteristics of SRS 10  
b) Write a short note on DFD 5

Section-III

- 5 a) What is requirement analysis? Explain briefly 10  
b) What is software architecture 5
- 6 a) What are the steps taken for avoiding, monitoring and managing risk 10

- |    |   |    |
|----|---|----|
|    | b) Write a short note on software project planning  | 5  |
| 7. | a) What is the relevance of design concept cohesion | 5  |
|    | b) Explain coupling, functional dependences         | 10 |

**Section -IV**

- |     |  |    |
|-----|--|----|
| 8.  | a) Explain the design concepts in software engineering | 10 |
|     | b) Differentiate between flowchart and structure chart | 5  |
| 9.  | a) Write a note on information hiding                  | 5  |
|     | b) Explain test driven development                     | 10 |
| 10. | a) Explain test case execution                         | 10 |
|     | b) Define boundary value analysis                      | 5  |